

**PATENT**

**Docket No. RSW920010045US1**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**INVENTORS:** Jeffrey Scott Chase and Ronald P. Doyle

**APPLICATION NO.** 09/894,016

**FILED:** June 28, 2001

**Examiner:** A. Boutah

**CASE NO.** RSW920010045US1

**Group Art Unit:** 2143

**TITLE:** METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT  
FOR HIERARCHICAL LOAD BALANCING

---

**FILED ELECTRONICALLY ON December 13, 2006**

---

**MAIL STOP APPEAL BRIEF-PATENTS**

**Commissioner for Patents**

**P.O. Box 1450**

**Alexandria, VA 22313-1450**

**Attention: Board of Patent Appeals and Interferences**

**APPELLANTS' BRIEF**

This Appeal Brief is in furtherance of the Notice of Appeal filed in this case on October 5, 2006. A pre-appeal conference decision was mailed on November 13, 2006, indicating the rejections of Claims 1-9 were withheld; thus, the due date for filing the Appeal Brief is December 13, 2006. The Commissioner is authorized to charge the fee for filing of this Appeal Brief to Deposit Account No. 09-0461.

**1. REAL PARTY IN INTEREST**

The present application is assigned to International Business Machines Corporation, having its principal place of business at New Orchard Road, Armonk, New York 10504. Accordingly, International Business Machines Corporation is the real party in interest.

**2. RELATED APPEALS AND INTERFERENCES**

The Appellants, assignee, and the legal representatives of both are unaware of any other appeal or interference which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

**3. STATUS OF CLAIMS**

- A. Claims canceled: None
- B. Claims withdrawn from consideration but not canceled: None
- C. Claims pending: 1-9
- D. Claims allowed: none
- E. Claims rejected: 1-9
- F. Claims appealed: 1-9

Appealed Claims 1-9 as currently pending are attached as the Claims Appendix hereto.

**4. STATUS OF AMENDMENTS**

A Request for Continued Examination under 37 C.F.R. §1.114 was filed on September 12, 2005 in which claim amendments were made. In response, the Examiner indicated the claim amendments were entered and issued a non-final Office Action on January 9, 2006. A Reply under 37 C.F.R. §1.112 was filed on April 10, 2006: no claim amendments were made. In response, the Examiner issued the final Office Action appealed herein.

**5. SUMMARY OF THE CLAIMED SUBJECT MATTER**

Claim 1: A system for improved load balancing in a client/server environment, comprising: at least one caching/hashing switch (CHS) coupled between clients and servers in said client/server environment (page 7, lines 2-7), said CHS storing previously-requested objects (page 7, lines 15-16), said CHS comprising: a hashing switch coupled to said servers (page 8, line 5); and a front end cache coupled between said clients and said hashing switch (page 8 lines 5-7); wherein object requests for objects stored in said CHS are satisfied immediately from said CHS (page 8, lines 8-13).

Claim 7: A method of improved load balancing in a client/server environment, comprising: receiving an object request from a client (page 9, line 10); determining if the object requested by said object request is stored in a cache coupled between said client

and a hashing switch coupled to a server farm (page 9, lines 10-15); if said object is stored in said cache, immediately returning a copy of said object to said client (page 9, lines 12-15); and if said object is not stored in said cache, then: hashing said object request using said hashing switch (page 9, lines 16-18); forwarding said hashed object request to said server farm (page 9, line 18); forwarding said requested object from said server farm to said cache for storage (page 9, lines 19-20); and returning a copy of said requested object to said client (page 9, line 21).

Claim 8: A computer program product for providing improved load balancing in a client/server environment, the computer program product comprising a computer usable medium having computer usable program code embodied therein, the computer usable program code comprising: computer usable program code configured to receive an object request from a client (page 9, line 10); computer usable program code configured to determine if the object requested by said object request is stored in a cache coupled between said client and a server farm (page 9, lines 10-15); computer usable program code configured to immediately return a copy of said object to said client if said object is stored in said cache (page 9, lines 12-15); computer usable program code configured to hash said object request (page 9, lines 16-18); computer usable program code configured to forward said hashed object request to said server farm (page 9, line 18); computer usable program code configured to forward said requested object from said server farm to said cache for storage (page 9, lines 19-20);

and computer usable program code configured to return a copy of said requested object to said client (page 9, line 21).

Claim 9: An improvement to a load balancing system in a client/server environment having at least one client coupled, via a network connection, to a plurality of servers, and a hashing switch coupled between said network connection and said plurality of servers, said improvement comprising: a cache coupled between said network connection and said hashing switch, said cache storing previously requested objects and configured to satisfy requests for said previously requested objects without passing said requests to said hashing switch (page 8, lines 5-18).

The present invention provides a system for improved load balancing in a client/server environment. At least one caching/hashing switch (CHS) is coupled between any clients and servers in the client/server environment. The CHS includes a hashing switch coupled to said servers for directing user requests and a front end cache coupled between said clients and said hashing switch used to store previously requested objects. If a user requests an object stored in the CHS (at the front end cache), the request is immediately satisfied from the copy stored in the front end cache.

**6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

Appellants request the Board to review the following rejections:

1. The rejection of Claims 1 and 9 under 35 U.S.C. §103(a) based on Applicants' Admitted Prior Art in view of U.S. Patent No. 6,006,264 to Colby et al.; and
2. The rejection of Claims 2-8 under 35 U.S.C. §103(a) based on Applicants' Admitted Prior Art in view of Colby and in further view of U.S. Patent No. 6,240,461 to Cielsak.

**7. ARGUMENTS**

**The Cited Art Does Not Render the Claims Obvious**

The Examiner Has Not Established a *Prima Facie* Case of Obviousness

As set forth in the MPEP:

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combined reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP 2143

- 1. The rejection of Claims 1 and 9 under 35 U.S.C. §103(a) based on Applicants' Admitted Prior Art in view of Colby.**

The present claimed invention focuses on improving load balancing in a server/client environment by utilizing a cache/hashing switch (CHS) coupled between clients and servers. Topographically, the present claimed invention places the hashing switch

between the clients and the servers, and places the cache between the hashing switch and the clients. By utilizing this configuration, any request for content sent by a client is first intercepted by the cache. The cache examines the request to determine if the content is cached locally, and only when the content is not cached is the request forwarded to the hashing switch for directing to the appropriate server. Specifically, claim 1 recites:

at least one caching/hashing switch (CHS) coupled between clients and servers in said client/server environment, said CHS storing previously requested objects, said CHS comprising:  
a hashing switch coupled to said servers; and  
a front end cache coupled between said clients and said hashing switch;  
wherein object requests for objects stored in said CHS are satisfied immediately from said CHS.

Each additional independent claim (Claims 7, 8 and 9) recites a form of these limitations. This configuration is neither taught nor suggested by the prior art of record, specifically the applicant admitted prior art (AAPA), Colby and/or Cieslak, taken alone or in combination.

The Examiner is essentially arguing that the mere existence of hashing switches and the mere existence of caches, both of which are acknowledged as being known, and the teaching of the content-aware flow switch of Colby, makes it obvious to combine them, and that such combination would achieve the present invention. This piecemeal approach is a clear application of hindsight; nothing in Colby even remotely suggests the placing of a front-end cache in front of a hashing switch, and immediately serving objects from the cache back to the client when they are stored in the cache. The Examiner has instead gone against the directive from the Federal Circuit in *In re Fritch* which states “[I]t is impermissible to use the claimed invention as an instruction manual or ‘template’ to piece

together the teachings of the prior art so that the claimed invention is rendered obvious....”  
977 F.2d 1260 (Fed.Cir. 1992).

Nothing in the references cited by the Examiner, nor in the AAPA, suggests the present claimed invention, where client requests arrive at the cache first and, if a request is available to be served from the cache (because it has been requested previously and is stored in the cache), it is served immediately back to the client without the need for any switching to occur. In the Final Office Action issued June 28, 2006, the Examiner asserts that Colby teaches a Content-Aware Flow Switch (CFS), which further includes a Content Server Database (CSD) for identifying whether the requested content is at a local server before performing switching function, e.g., redirecting, the request to a remote server. The Examiner additionally asserts that the functionality of the CSD is equivalent to immediately serving the client with the requested content.

Applicants respectfully disagrees with Examiner’s interpretation of Colby. Colby specifically defines the CSD as “containing information about content flow characteristics, content locality, and the location of and the load on servers” (column 6, lines 42-44). Nowhere in Colby does it suggest that the CSD functions as a cache for storing previously requested content, nor does Colby disclose or reasonably suggest providing content to a requesting client directly from the CSD.

Column 7, lines 20-52 of Colby illustrate an example of how the CSD handles a content request. Generally, the CSD examines its records of servers and the content each server has stored therein, and generates a listing of available servers that contain the requested content for delivery to a switching element. The switching element will use the



list of servers to connect the requesting client with an available server. However, no mention is made in Colby of utilizing a cache between a client and a hashing switch, and providing requested content from the cache if the content is stored locally at the cache. Additionally, no mention is made in Colby relating to storing client requested content at the CSD as is asserted by the Examiner. Colby defines the CSD as a database containing information relating to flow characteristics, content locality, and the location of and load on individual servers. The CSD uses this information to instruct the switch to direct a user request to an individual server that can satisfy the request, but the CSD itself never satisfies a user request for content itself as is specifically claimed in the present invention.

The Examiner has failed to show the necessary suggestion in the prior art to modify the teachings to achieved the present claimed invention. Even if the teachings and suggestions of the cited references *were* combined, they still would not result in the claimed invention, since there would never be a configuration where the cache is situated between the client and the hashing switch.

Each of the pending claims specifically recites the novel and non-obvious arrangement set forth above. None of the cited prior art teach these novel features, and none of the cited prior art contains any suggestion of such a combination. Without such a suggestion, it is inappropriate to reject the claims as being obvious based on the cited prior art.

**2. The rejection of Claims 2-8 under 35 U.S.C. §103(a) based on Applicants' Admitted Prior Art in view of Colby and in further view of Cielsak.**

As discussed above, the combination of AAPA and Colby fails to teach or reasonably suggest the present claimed invention. The addition of Cieslak fails to identify or address the problem solved by the present invention, that is, minimizing the burden on a hashing switch and a server farm. In the present invention, if an object is cached, it is immediately served back to the client, and thus the object does not have to be re-hashed and the server does not have to re-serve it. Additionally, Cieslak fails to suggest modifying its teachings to arrive at the claimed invention.

As before, the Examiner has failed to show the necessary suggestion in the prior art to modify the teachings to achieved the present claimed invention. Even if the teachings and suggestions of the cited references *were* combined, they still would not result in the claimed invention, since there would never be a configuration where the cache is situated between the client and the hashing switch.


Each of the pending claims specifically recites the novel and non-obvious arrangement set forth above. None of the cited prior art teach these novel features, and none of the cited prior art contains any suggestion of such a combination. Without such a suggestion, it is inappropriate to reject the claims as being obvious based on the cited prior art.

**8. CONCLUSION**

For the foregoing reasons applicants respectfully request this Board to overrule the Examiner's rejections and allow Claims 1-9.

Respectfully submitted,

December 13, 2006  
Date

  
\_\_\_\_\_  
John R. Brancolini, Reg. No. 57,218  
Synnestvedt & Lechner LLP  
1101 Market Street  
Suite 2600  
Philadelphia, PA 19107  
Telephone: 215-923-4466  
Facsimile: 215-923-2189

**CLAIMS APPENDIX**

**CLAIMS INVOLVED IN THIS APPEAL:**

1. (Previously presented) A system for improved load balancing in a client/server environment, comprising:

at least one caching/ hashing switch (CHS) coupled between clients and servers in said client/server environment, said CHS storing previously-requested objects, said CHS comprising:

a hashing switch coupled to said servers; and

a front end cache coupled between said clients and said hashing switch;

wherein object requests for objects stored in said CHS are satisfied immediately from said CHS.

2. (Original) The systems of claim 1, wherein said CHS also hashes object requests, and wherein:

object requests which are not stored in said CHS are hashed;

each of said hashed object requests are forwarded to a respective server on which each requested object is stored;

each of said requested objects is forwarded to said CHS and stored thereon; and

a copy of each of said requested objects is returned to a respective client requesting said object.

3. (Original) The system of claim 2, wherein said objects are web objects and wherein said CHS comprises:

a web proxy cache; and

a URL-hashing switch coupled to said web proxy cache.

4. (Previously presented) The system of claim 2, wherein said objects are web objects and wherein said CHS comprises:

computer usable code configured to operate as a web proxy cache for storing retrieved web objects; and

computer usable code configured to operate as a URL-hashing switch, for hashing said web object requests and forwarding said hashed web object requests to said respective servers.

5. (Original) The system of claim 4, wherein said client/server environment comprises a plurality of clients coupled to at least one server farm via a network connection.

6. (Original) The system of claim 4, wherein said client server environment comprises a plurality of clients coupled to a plurality of server farms via a network connection, and wherein each of said server farms has a CHS associated therewith, and wherein said system further comprises:

a Level 4 switch coupled between said network connection and said CHS's.

7. (Previously presented) A method of improved load balancing in a client/server environment, comprising:

receiving an object request from a client;

determining if the object requested by said object request is stored in a cache coupled between said client and a hashing switch coupled to a server farm;

if said object is stored in said cache, immediately returning a copy of said object to said client; and

if said object is not stored in said cache, then:

hashing said object request using said hashing switch;

forwarding said hashed object request to said server farm;

forwarding said requested object from said server farm to said cache for storage; and

returning a copy of said requested object to said client.

8. (Previously presented) A computer program product for providing improved load balancing in a client/server environment, the computer program product comprising a computer usable medium having computer usable program code embodied therein, the computer usable program code comprising:

computer usable program code configured to receive an object request from a client;

computer usable program code configured to determine if the object requested

by said object request is stored in a cache coupled between said client and a server farm;

computer usable program code configured to immediately return a copy of said object to said client if said object is stored in said cache;

computer usable program code configured to hash said object request;

computer usable program code configured to forward said hashed object request to said server farm;

computer usable program code configured to forward said requested object from said server farm to said cache for storage; and

computer usable program code configured to return a copy of said requested object to said client.

9. (Previously presented) An improvement to a load balancing system in a client/server environment having at least one client coupled, via a network connection, to a plurality of servers, and a hashing switch coupled between said network connection and said plurality of servers, said improvement comprising:

a cache coupled between said network connection and said hashing switch, said cache storing previously requested objects and configured to satisfy requests for said previously requested objects without passing said requests to said hashing switch.

**EVIDENCE APPENDIX**

No additional evidence is presented.



**RELATED PROCEEDINGS APPENDIX**

No related proceedings are presented.